IN THE CLAIMS

Please cancel claim 1 as follows:

1. (CANCELLED)

2. (PREVIOUSLY PRESENTED) A digital data processing method comprising

responding to a request for access by at least initiating transfer of data designated by that request over a communications pathway that does not include a node that controls such access;

the initiating step including at least initiating transfer of the data using administrative information obtained from the node;

wherein responding to the request comprises a bypass, interceding in response to the request for access applied thereby to the node that controls such access.

3. (PREVIOUSLY PRESENTED) A digital data processing method comprising

responding to a request for access to a peripheral device by at least initiating transfer of data designated by that request over a communications pathway that does not include a node that controls access to the peripheral device;

the initiating step including at least initiating transfer of the data using administrative information obtained from the node pertaining to storage of data on the peripheral device;

wherein responding to the request comprises a bypass, interceding in response to the request for access to the peripheral device applied thereby to the node that controls such access to the peripheral device.

4. (PREVIOUSLY PRESENTED) A digital data processing method comprising

responding to a request ("first access request") by a first node for access to a peripheral device by at least initiating transfer of data designated by that request between the first node and the peripheral device over a communications pathway that does not include a second node that controls access by at least the first node to the peripheral device;

the initiating step including at least initiating transfer of the data using administrative information obtained from the second node pertaining to storage of data on the peripheral device;

wherein responding to the request comprises a bypass, interceding in response to the request for access by the first node to the peripheral device applied thereby to the second node that controls such access to the peripheral device.

5. (ORIGINAL) The method of claim 4, wherein the first node is a client node and the second node is a server node.

6. (ORIGINAL) The method of claim 4, wherein the initiating step includes at least initiating transfer of the data designated by the first access request between the first node and the peripheral device over a communications pathway that differs from that over which a file system executing on at least one of the first and second nodes would transfer data in response to the first access request.

7. (ORIGINAL) The method of claim 4, wherein the initiating step includes at least initiating transfer of the data designated by the first access request between the first node and the peripheral device via a directly attached disk connect.

8. (ORIGINAL) The method of claim 4, wherein the initiating step includes at least initiating transfer of the data designated by the first access request between the first node and the peripheral device via fibre channel.

9. (ORIGINAL) The method of claim 4, wherein the initiating step includes at least initiating transfer of the data designated by the request via a direct connection between the first node and the peripheral device, wherein the direct connection comprises any of a fibre channel, a firewire bus, a serial storage architecture (SSA) bus, a high-speed Ethernet bus, a high performance parallel interface (HPPI) bus, and other high-speed peripheral device bus.

10. (ORIGINAL) The method of claim 6, wherein the initiating step includes at least initiating transfer of the data designated by the first access request to physical storage locations on the peripheral device determined from the administrative information.

11. (ORIGINAL) The method of claim 10, comprising obtaining as administrative information a map indicative of physical locations at which the data are stored on the peripheral device.

12. (ORIGINAL) The method of claim 11, wherein the responding step comprises obtaining the map by generating and applying to the file system a second access request.

13. (ORIGINAL) The method of claim 12, wherein the second access request is for access to a logical unit to which access is controlled by the second node.

14. (ORIGINAL) The method of claim 13, wherein the second request is for access to a logical unit other than a file designated in the first access request.

15. (ORIGINAL) The method of claim 14, wherein the second access request is a request to write a file.

16. (ORIGINAL) A digital data processing method of accessing a peripheral device, the method comprising

responding to a request ("first access request") by a first node for access to the peripheral device by bypassing, at least in part, a file system that would otherwise respond to the first access request by transferring data designed thereby the between the first node and the peripheral device over a communications pathway that includes a second node that controls access by the first node and one or more other nodes to the peripheral device;

the responding step including at least initiating transfer of data designated by the first access request between the first node and the peripheral device over a communications pathway that does not include the second node;

the initiating step including at least initiating transfer of the data using administrative information obtained from the second node pertaining to storage of data on the peripheral device.

17. (ORIGINAL) The method of claim 16, wherein the initiating step includes at least initiating transfer of the data designated by the first access request to physical storage locations on the peripheral device determined from the administrative information.

18. (ORIGINAL) The method of claim 17, wherein the responding step comprises obtaining as administrative information a map indicative of physical locations at which the data are stored on the peripheral device.

19. (ORIGINAL) A digital data processing method accessing a peripheral device, the method comprising

intercepting, with a filter driver executing on a first node, a request ("first access request") by that node for access to the peripheral device,

at least initiating transfer of the data designated by that request between the first node and the peripheral device using file mapping obtained from a second node.

20. (ORIGINAL) The method of claim 19, comprising obtaining the file mapping by sending a request ("second access request") to the second node for access to a logical unit other than a file designated in the first access request.

21. (ORIGINAL) The method of claim 20, wherein the logical unit is a file that resides on a peripheral device local to the first node.

22. (ORIGINAL) The method of claim 20, wherein the second access request is a request to write a file.

23. (ORIGINAL) The method of claim 20, comprising generating and applying to the file system a third access request, the third request designating the logical unit designated by the second access request.

24. (ORIGINAL) The method of claim 19, wherein the first node is a client node and the second node is a server node.

25. (ORIGINAL) The method of claim 19, wherein the initiating step includes at least initiating transfer of data designated by the first access request between the first node and the peripheral device over a communications pathway that differs from that over which a file system executing on at least one of the first and second nodes would transfer data in response to the first access request.

26. (ORIGINAL) The method of claim 19, wherein the initiating step includes at least initiating transfer of data designated by the first access request between the first node and the peripheral device via a directly attached disk connect.

27. (ORIGINAL) The method of claim 19, wherein the initiating step includes at least initiating transfer of data designated by the request via a direct connection between the first node and the peripheral device, wherein the direct connection comprises any of a fibre channel, a firewire bus, a serial storage architecture (SSA) bus, a high-speed Ethernet bus, a high performance parallel interface (HPPI) bus, and other high-speed peripheral device bus.

28. (PREVIOUSLY PRESENTED) A bypass for use in a first node to improve access to information stored on a peripheral device, the bypass comprising

a first filter driver executing within and coupled to a file system of the first node, the filter driver responding to a request ("first access request") generated within the first node for access to the peripheral device by at least initiating transfer of data designated by that request between the first node and the peripheral device using administrative information obtained from a second node that controls access by the first node and one or more other nodes to the peripheral device,

the filter driver at least initiating the transfer via a communications pathway that does not include the second node;

wherein responding to the request comprises a bypass, interceding in response to the request for access by the first node to the peripheral device applied thereby to the second node that controls access to the peripheral device.

29. (ORIGINAL) The bypass of claim 28, wherein the first filter driver wherein obtains as administrative information a map indicative of physical locations at which the data are stored on the peripheral device.

30. (ORIGINAL) The bypass of claim 28, wherein the first filter driver obtains the map by generating a request ("second access request") to the second node for access to a logical unit other than a file designated in the first access request.

31. (ORIGINAL) The bypass of claim 30, wherein the logical unit is a file that resides on a peripheral device local to the first node.

32. (ORIGINAL) The bypass of claim 31, wherein the second access request is a request to write a file.

33. (ORIGINAL) The bypass of claim 29, comprising a second filter driver executing within and coupled to a file system of the second node.

34. (ORIGINAL) The bypass of claim 33, wherein the second filter driver generates the map in response to a further access request generated by the first filter driver.

35. (ORIGINAL) The bypass of claim 28, wherein the first filter driver selectively issues a request to the file system of the first node to create a mapped device corresponding to the peripheral device.

36. (ORIGINAL) The bypass of claim 35, wherein the first filter driver compares information regarding a network volume being mounted to local volumes to which the first filter driver has access to determine whether a communications pathway exists for transfer of data between the first node and the peripheral device that does not include the second node.

37. (ORIGINAL) The bypass of claim 36, wherein the communications pathway over which the first filter driver transfers data designated by the first access request differs from that over which a file system executing on at least one of the first node and the second node would transfer data in response to the first access request.

38. (ORIGINAL) The bypass of claim 36, wherein the communications pathway over which the first filter driver transfers data designated by the first access request is a direct connection.

39. (ORIGINAL) The bypass of claim 38, wherein communications pathway over which the first filter driver transfers data designated by the first access request comprises a fibre channel.

40. (ORIGINAL) The bypass of claim 38, communications pathway over which the first filter driver transfers data designated by the first access request comprises any of a fibre channel, a firewire bus, a serial storage architecture (SSA) bus, high-speed Ethernet bus, high performance parallel interface (HPPI) bus, and other high-speed peripheral device bus.